future internet

MDPI

Article

Data Anonymization for Hiding Personal Tendency in Set-Valued Database Publication

Dedi Gunawan [1,*] and Masahiro Mambo [2]

[1] Division of Electrical Engineering and Computer Science, Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Ishikawa 920-1192, Japan

[2] Faculty of Electrical and Computer Engineering, Institute of Science and Engineering, Kanazawa University, Kanazawa, Ishikawa 920-1192, Japan; mambo@ec.t.kanazawa-u.ac.jp

* Correspondence: dedigun@stu.kanazawa-u.ac.jp

Received: 1 June 2019; Accepted: 18 June 2019; Published: 20 June 2019

Abstract: Set-valued database publication has been increasing its importance recently due to its benefit for various applications such as marketing analysis and advertising. However, publishing a raw set-valued database may cause individual privacy breach such as the leakage of sensitive information like personal tendencies when data recipients perform data analysis. Even though imposing data anonymization methods such as suppression-based methods and random data swapping methods to such a database can successfully hide personal tendency, it induces item loss from records and causes significant distortion in record structure that degrades database utility. To avoid the problems, we proposed a method based on swapping technique where an individual's items in a record are swapped to items of the other record. Our swapping technique is distinct from existing one called random data swapping which yields much structure distortion. Even though the technique results in inaccuracy at a record level, it can preserve every single item in a database from loss. Thus, data recipients may obtain all the item information in an anonymized database. In addition, by carefully selecting a pair of records for item swapping, we can avoid excessive record structure distortion that leads to alter database content immensely. More importantly, such a strategy allows one to successfully hide personal tendency without sacrificing a lot of database utility.

Keywords: set-valued database; data anonymization; hiding personal tendency; swapping technique

1. Introduction

Publishing a set-valued database such as a web access database brings many benefits for various applications like market analysis, advertising and recommender systems. Table 1 shows an example of such a database containing URL's of individual's web access as items. To simplify discussion, each URL is expressed by number as its ID. Releasing such a database to the public imposes some challenges since data recipients can directly associate an individual with his/her record to infer personal sensitive information such as personal preference or personal tendency. According to the Oxford on-line dictionary the term tendency refers to an inclination towards certain characteristics or type of behavior. In this context, we consider that individual tendency is an inclination of someone toward certain groups or category of items. For example, a person who accesses to some specific website such as football.com, espn.com, and livescore.com has a tendency to football. Therefore, the more someone accesses to a certain category of websites, the more obvious that the person has a tendency to the category. For some people, their personal tendency toward certain things can be considered as sensitive information and there is a trade-off for disclosing it [1].

Hiding personal tendency in a set-valued database i.e., web access log is essential for security application such as to avoid the Internet users get email spam and phishing. To illustrate the importance

of hiding personal tendency we give the following example. An internet user, Bob, accessed various websites where most of the websites that he visited can be categorized as automobile websites, and to this end we can say that Bob has a strong tendency to the automobile category. The web address that Bob accessed are collected by On-line advertisement company, says AdOn. In one day AdOn release the log of the internet users, including Bob's web access log to its collaborator company, namely AdOff. Unfortunately, an irresponsible agent who is working in AdOff knows the information about Bob's tendency by analyzing his browsing history and starts to send email spam that contains interesting information about automobiles, as well as inserting a malicious link in the email, due to his curiosity Bob may get in trouble if he clicks the malicious link.

The illustration story tells us that the spammer sends email spam that is precisely related to the targeted personal tendency. However, if Bob's tendency is hidden, the spammer might fail to trap Bob, since the email spam would not be too precise to Bob's personal tendency. As a result, even if Bob receives the email he might ignore it and refrain to click a link that does not attract him.

As the nature of a set-valued database where an individual is directly correlated to his/her items, accordingly, prior to publishing such a database to the public, one should consider a data anonymization process such that the personal tendency of individuals can be protected in a published database. Several data anonymization techniques such as generalization based techniques, suppression based techniques, substitution based technique and swapping based techniques have been intensively studied and each of them has pros and cons. However, there are very few researches on hiding personal tendency in set-valued database.

Employing data anonymization techniques by means of a generalization based approach such as k-anonymity [2] and k^m-anonymity [3] cannot successfully hide personal tendency even if some items are changed to its general value. Such a failure occurs due to generalization only replaces specific item with its general value i.e., category which still pictures the tendency of an individual. For example, Bob likes to access autocars.net which is categorized as an automobile website, but replacing the website name with its category i.e., automobile, it does not hide that Bob has a tendency toward automobiles. In addition, employing generalization technique such as full domain generalization in set-valued database leads significant reduction in data utility [4] and causes item loss in databases if several items from the same category appear together in one record [5]. As a result, data recipients cannot obtain maximum utility from the database.

Suppression based data anonymization techniques [4,6,7] can be more realistic for hiding personal tendency since certain items in a record are removed so that there is no clue to find the suppressed items. Unfortunately, such a strategy may cause significant distortion in record structure and item loss in the database, making an anonymized database becomes less useful for data recipients. Here record structure refers to the composition or a set of items that construct the records in a database.

The swapping technique moves values or items from a record to another record. The advantage is that it does not reduce the number of items as well as does not cause item loss in a database. Accordingly, we seek a solution to develop an anonymization algorithm based on swapping technique in order to overcome drawbacks of other techniques described above. Moreover, it can hide personal tendency by deeming a personal tendency as a tendency of a different individual. As a result, the individual's true tendency is hidden and cannot be easily identified.

In this paper, we propose an algorithm that ensures to hide personal tendency while at the same time it does not excessively distort the record structure of an individual. To achieve these properties, we only select a category that has the highest item frequency in a record and swap its items to another record that has a minimum similarity.

The rest of the paper is organized as follows. Section 2 describes several works in protection of individual information and current development of swapping technique. Sections 3 and 4 discuss the problem that we are going to solve and our proposed method, respectively. Lastly, Sections 5 and 6 provide experimental results and conclusion, respectively.

Table 1. An example of visited URL's.

TID	UID	IID
1	001	2 3 8 10 11
2	002	2 7 8 10
3	003	5 6 7 10 12
4	004	4 6 7 9 10 12
5	005	1 3 5 8 11
6	006	2 3 8 9
7	007	1 4 6 7 9 12
8	008	3 4 7 8 11
9	009	2 6 9 12
10	010	1 3 5 9 10

2. Related Work

There have been many data anonymization techniques that are proposed to protect personal sensitive information in relational databases and micro datasets [8,9]. While the research in set-valued database anonymization mainly focuses on de-associating records or sensitive items to their data subject. In contrast, there are relatively few methods proposed to protect personal tendency in set-valued databases.

2.1. Protection of Personal Sensitive Information and Personal Tendency

Countless number of methods were proposed to protect personal sensitive information and personal tendency in data publishing scenario. A method to limit privacy breaches on transaction database has been demonstrated in [10]. It can employ pseudorandom generator to construct randomized transactions. The seed of the pseudorandom generator can also work to compress randomization transactions such that the magnitude of randomized transactions does not exceed that of original transactions.

A different privacy protection method in distributed database scenario has also been proposed in [11]. The method called (I, A)-privacy constraint ensures that user's queries in his intention I are hidden from a set A of adversarial principles, i.e., servers and other users due to the restriction mechanism of the relational database. In addition, it is stated in [11] that the implementation of such a method is incredibly time-consuming since it requires the users to enumerate all possible query plans.

Split personality is proposed in [12] to protect personal tendency where the log of users' queries are split based on their interests so as to result in some multiple personalities. For instance, a user who input queries related to Traveling and on-line shopping categories can be split into two distinct individuals with different personalities, one for the queries about traveling and the other one for the queries about on-line shopping. Even tough the method does not cause item loss from a database, this method distorts the correlation among the items in the record and thus the record structure is drastically changed. Furthermore, such an approach causes a database size to be much larger that leads to degrading much database utility.

The ρ-uncertainty proposed in [7] utilizes partial item suppression based on a heuristic approach which removes only items in sensitive association rules (SAR) to avoid excessive information loss. Even though the method can successfully hide SAR under the parameter ρ, it may cause item loss in an anonymized database. Accordingly, data recipient cannot obtain maximum utility. We present the comparison of our algorithm with others in Table 2.

Table 2. Comparison of methods protecting personal sensitive information and personal tendency.

Method	Database Type	Technique	Record Structure	No. of Records
(I, A)-privacy [11]	Relational	Access control query	Preserved	Kept
Amplification [10]	Set-valued	Randomization and noise addition	Changed	Increase
Split personality [12]	Set-valued	Splitting record	Changed	Increase
ρ-uncertainty [7]	Set-valued	Suppressing items	Changed	Decrease
Rank swapping [13]	Microdata	Random based swapping	Changed	Kept
Proposed method	Set-valued	Similarity based swapping	Changed	Kept

2.2. Swapping Techniques

Data swapping techniques have been widely used for statistical disclosure control in micro dataset publishing. Even though there is a dispute of its side effect, i.e., the techniques induce information inaccuracy at a record level due to some values of records are swapped to another record. However, such techniques can preserve items in a set-valued database from loss. As a result, data recipients may obtain all information of the items even if the database has been anonymized. In addition, in a set-valued database, the side effect can also bring advantage for certain application like recommender system. When items of an individual are swapped with items from another individual, there is a chance for the individuals to view different things in his/her recommendation list, as a result, the recommender system can increase its serendipity. In recommender system serendipity, it is important to overcome overspecialization problem and broaden user preferences [14].

The pioneering work on data anonymization using swapping technique [9,13] has successfully implemented it to micro datasets for anonymizing numerical and categorical attributes. Another popular swapping method to protect privacy in a micro dataset database is called rank swapping. The method firstly sorts all the values of attribute X in ascending order, then each value $v \in X$ is swapped randomly with other value $v' \in X$ under the defined range p% of the number v. It is stated in [15] that rank swapping results in better statistical data utility than ordinary data swapping.

Motivated to improve rank swapping, two protection schemes namely rank swapping p-distribution and rank swapping p-buckets are proposed in [16]. The former combines normal probability distribution of the database with defined p% value to swap value $v \in X$ with other value $v' \in X$. On the other hand, rank swapping p-buckets splits the sorted values of an attribute 'X' into several p buckets. It firstly selects a bucket B_s based on a certain probability function and then the values in B_s are swapped randomly and uniformly.

Those previously mentioned swapping methods work in micro datasets where in general it only contains a small number of data attributes and therefore it cannot be directly implemented in set-valued databases which usually have a large number of attributes. Moreover, the existing rank swapping techniques rely on randomization to swap the values which cause severe distortion record structure in set-valued databases.

A more recent swapping technique has been implemented to avoid negative association rules [17]. It firstly creates databases to satisfy ℓ-diversity [18]. Once the databases satisfy that, the next step is to find all records containing negative association rules and put the records in set Y. Sensitive values in negative association rules that are not compatible are swapped to another record that is compatible with the sensitive attribute in set C. These steps are recursively run until there is no intersection between Y and C.

3. Problem Formulation

A set-valued database $\mathcal{D}$ consists of n records t_k for $k = 1$ to n. Each t_k is composed of record ID TID, user ID UID and a set IID of item ID's i obtained from $I = \{i_1, i_2, i_d, .., i_{|I|}\}$, such that each individual is directly associated with his/her items. Consequently, if such a database is published plainly, malicious data recipients can conduct some data analysis to infer sensitive information such as personal tendency of any individual in $\mathcal{D}$. Note that only IID among the components of t_k may be

processed during the anonymization. In such a case, we simply use t_k as a set of items, e.g., t_3 = {5, 6, 7, 10, 12} in Table 1, if it does not cause confusion.

In real life a person tends to have several preferences, for example, the person may like travelling, on-line shopping or culinary. In our problem formulation, a database owner who publishes his/her database specifies such preference as category. A set of all such categories is denoted by G. Each item in the databases is assigned to one of these categories (see Figure 1). Then we can define personal tendency as follows.

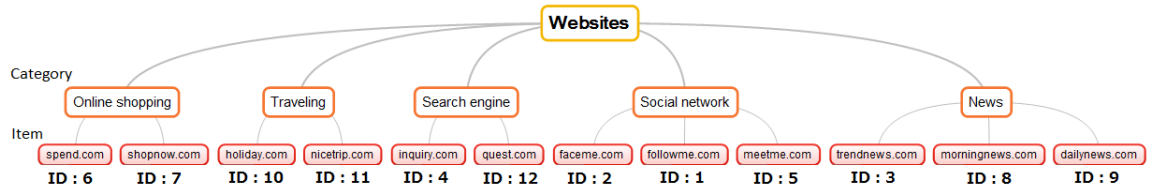


Figure 1. Website category.

Definition 1. *Personal tendency is a set of categories of items existing in the person's record. Each category has a weight that is the number of items belonging to the category in a record.*

In real situations, given a set-valued database processing items directly is more practical than handling categories in data analysis. Thus, we give another definition as follows.

Definition 2. *Personal tendency is a set of items belonging to a certain category that exists in a record.*

To hide personal tendency, we need a data anonymization method like swapping. The swapping refers to selecting items from a record as an itemset and swap the itemset to that of another record. In the case of random swapping, the items are selected randomly, while in our proposed scheme, we select items based on a certain procedure and consider them as an itemset for swapping.

Definition 3. *A swapping method imposed on the database $\mathcal{D}$ with n number of t to generate a swapped database $\widetilde{\mathcal{D}}$ is said to hide personal tendency if items of a category in a record are swapped to other items of different category in another record in a way such that it cannot be associated to its original data subject.*

To avoid excessive distortion of record structure, swapped items should be carefully selected. We give the following definition of hiding personal tendency corresponding to the case that items from a category with the highest frequency are selected.

Definition 4. *A swapping method imposed to database $\mathcal{D}$ with n number of t to generate a swapped database $\widetilde{\mathcal{D}}$ is said to hide personal tendency with a modest distortion if items of a category that has the highest frequency are swapped to another record in a way such that it cannot be associated to its original data subject.*

As an example, based on Figure 1 and Table 1, we can see that a user with UID = 003 has a more tendency toward On-line shopping since the on-line shopping category has two items with IID = {6, 7} while the other category has only one item. Therefore, to avoid excessive record structure distortion, we only swap items from such a category to another record.

3.1. Limitation of Traditional Rank Swapping

Employing traditional rank swapping in set-valued databases is not encouraged since it relies on random swapping of items, which causes significant distortion of record structure. When the structure of records drastically changes, database utility will also be significantly affected specifically for data

mining analysis like frequent itemset mining. Moreover, random swapping does not guarantee to result in true swap, that is, items in a record that belong to a specific category are swapped with other items from a different category in another record.

As an example, in the Table 1 we can inspect that UID = 008 has a tendency toward News. If we aim to hide the personal tendency of UID = 008 by swapping its items that belong to News category, i.e., IID = {3, 8} to other items from another record such as UID = 003 which has tendency toward Online shopping since it has IID = {6, 7}, such a swapping can hide the personal tendency of UID = 008 since the swapped items belong to different categories. Therefore, if we rely on a random swapping technique, there is no guarantee to achieve a true swap since there is a possibility that items from the same category are swapped.

3.2. Personal Tendency Similarity

In a database which has several records, an individual record might have a certain degree of similarity with others' records. This similarity coefficient can be exploited to ensure that the personal tendency can be hidden successfully. Prior to performing the swapping technique, we calculate the Jaccard similarity coefficient, *jc*, which measures the similarity between two records t_k and t'_k in a database by dividing their intersection items from their union items.

$$jc(t_k, t'_k) = \frac{(t_k \cap t'_k)}{(t_k \cup t'_k)}. \quad (1)$$

The *jc* of each pair of t_k and t'_k falls between $0 \leq jc(t_k, t'_k) \leq 1$. The smaller the *jc* of a pair, the smaller the similarity between them and vice versa. As mentioned in [12], 0.5 or higher is the most suitable Jaccard coefficient value to find high similarity. Intuitively, when items in a record are swapped with items from another record which has high similarity, the personal tendency of an individual is not hidden, since both have almost the same items. Thus, selecting the record which results in the minimum *jc* value can successfully hide personal tendency. By doing so, the tendency of a person is deemed to be the tendency of another person who has different characteristics. Another advantage of selecting records that result in the minimum *jc* is to avoid item collision that reduces the number of items in a dataset. It further impacts on data utility in an anonymized dataset.

3.3. Database Utility

Data utility of an anonymized database should also be taken into account since data recipients conduct further data analysis. We should note that every data anonymization results in anonymized databases that have lower data utility than that of the original one. Therefore, the issues in data anonymization are not only ensuring privacy protection in a published database but also preserving data utility after performing data anonymization algorithm.

There are various metrics to measure both issues in literature. Unfortunately, since there are no generic data utility measurements [19], we evaluate the anonymized database utility by using typical applications of data analysis in set-valued databases such as frequent itemset mining [20]. Therefore, we perform frequent itemset mining tasks over original and anonymized databases using the FP-Growth algorithm [21].

To quantify the amount of preserved data utility, one often uses frequent itemset FI that is the itemset appearing in many records of a database. In this case we count the number of frequent itemsets FI from the anonymized databases resulted by our proposed method and other existing methods, as well as that of the original databases. Since swapping items causes information inaccuracy at the record level, we further investigate the effect of the swapping to the information inaccuracy at the record level by evaluating the similarity of frequent itemset mining results using Equation (2). $F_{\mathcal{D}}$ and $F_{\widetilde{\mathcal{D}}}$ refer to the set of FI in the original database $\mathcal{D}$ and that in the anonymized database $\widetilde{\mathcal{D}}$, respectively. When the information inaccuracy at the record level is low, the similarity of the mining result is high. Therefore, the higher the similarity is, the better the dataset utility is preserved in the anonymized databases.

$$U(\widetilde{\mathcal{D}}, \mathcal{D}) = \frac{|F_{\mathcal{D}} \cap F_{\widetilde{\mathcal{D}}}|}{|F_{\mathcal{D}} \cup F_{\widetilde{\mathcal{D}}}|}. \quad (2)$$

3.4. Database Dissimilarity

Database dissimilarity is another side effect of transforming a database into an anonymized one. The database dissimilarity refers to the difference of database content between the original and the anonymized one. To measure the similarity between an original database and anonymized database, we can apply dissimilarity metric *Diss* [22]. $f\mathcal{D}$ and $f\widetilde{\mathcal{D}}$ represent the number of occurrences of item i in original database and that in anonymized database respectively, while d and $\widetilde{d}$ are the total number of distinct items in $\mathcal{D}$ and $\widetilde{\mathcal{D}}$, respectively.

$$Diss(\mathcal{D}, \widetilde{\mathcal{D}}) = \frac{1}{\sum_{i=1}^{d} f\mathcal{D}(i)} \times \left| \sum_{i=1}^{d} f\mathcal{D}(i) - \sum_{i=1}^{\widetilde{d}} f\widetilde{\mathcal{D}}(i) \right|. \quad (3)$$

In the swapping techniques, the dissimilarity occurs mainly because of items collision. The collision emerges if items that are going to be swapped already exist in another record that has been determined as destination record. Therefore, the content of an anonymized database differs from that of the original one. Our solution to avoid excessive dissimilarity is by selecting a pair of records that has significant items difference for items swapping, such that the items collision can be minimized.

4. Proposed Method

The proposed scheme aims to guarantee the hiding of personal tendency by changing a person's tendency closer to another person's tendency while minimizing record structure distortion. To achieve that goal, in this scheme, we first measure the weight of each category based on the frequency of occurrence of items in the same category in a record. The more frequent items within the same category appear in a record, the stronger the tendency of an individual to that category is. We propose two schemes namely partial swapping and full swapping. Refer to Figure 2 to see an illustration of the proposed schemes. At first, the partial swapping is described from Sections 4.1 to 4.3. Then the detail of full swapping is described in Section 4.4. In the following discussion personal tendency refers to the Definition 1 except the case that we clearly state it refers to the Definition 2.

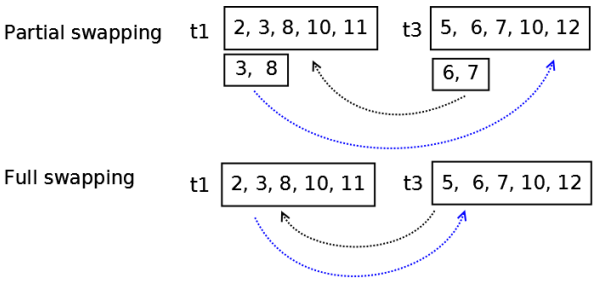


Figure 2. Swapping strategy.

4.1. Weighting Individual Tendency

In real life, an individual may have a certain tendency to specific categories, for example in Table 1, Alice who has UID = 004 accesses several websites that belong to some categories such as news, travelling and on-line shopping. However, among the sites that she accessed, traveling category occupies a higher portion than others. Thus, we can say that Alice has a tendency to travelling since it has the highest weight. The weight of a category refers to the number of items belonging to the category in a record.

The number of items in t_k is the length $|t_k|$ of the record and a website category is denoted by g (refer to Figure 1 as an example). For each t_k weight ratio WR_{t_k,g_j} of each category is computed as shown by (4), where g_j is the j-th category in the t_k.

$$WR_{t_k,g_j} = |\{i | i \in t_k \wedge i \in g_j\}|. \tag{4}$$

4.2. Item Selection for Partial Swapping

The concept of swapping in this proposal is different from the existing swapping techniques. At first, we count the frequency of items in each $t_k \in \mathcal{D}$ based on its category. Thus, we can understand the magnitude of an individual tendency from his/her record.

It is undoubtedly true that randomly swapping items from a record to another record can successfully hide personal tendency. However, such an operation may reduce usefulness of the modified database. To avoid such a negative effect, we introduce our heuristic solution called mean tendency value (MTV) which acts as a threshold. The MTV_{t_k} is determined for each t_k by dividing the record's length $|t_k|$ over the number of categories g_j whose items appear in the t_k, as shown in Equation (5). For each g_j which has weight higher than MTV_{t_k}, add all the g_j into a bucket Cg. After that, we select the g_j with the highest weight out of Cg as g'_j and obtain all the items of the selected $g'_j \in t_k$ as the items for swapping. Such procedures of item selection is shown in Algorithm 1 which outputs a set of $I_{swap} = \{I_{t_k} | t_k \in \mathcal{D}\}$ where $I_{t_k} = \{i | i \in t_k \wedge i \in g'_j\}$.

However, if all the WR_{t_k,g_j} in t_k have equal value with MTV_{t_k}, we do not swap any of its items to the other records since there is no significant information to learn about the individual that has an equal tendency to all g_j.

$$MTV_{t_k} = \frac{|t_k|}{|\{g_j \in G | i \in t_k \wedge i \in g_j\}|}, \tag{5}$$

where G is a set of all categories of the database as explained in Section 3.

Algorithm 1: Procedure of item selection.

INPUT : $\mathcal{D}$;
OUTPUT : I_{swap} ; // I_{swap} is a set of items from selected g'_j
Read $\mathcal{D}$;
Set C_g as an empty set;
foreach $t_k \in \mathcal{D}$ **do**
  compute the $|t_k|$;
  compute the MTV_{t_k};
  foreach $g_j \in t_k$ **do**
    if $WR_{t_k,g_j} > MTV_{t_k}$ **then**
      append g_j to Cg;
    end
  end
  $g'_j \leftarrow g_j$ having the maximum weight out of C_g;
  select $i \in g'_j$ as items for swapping;
  clear the C_g;
end

4.3. Destination Record Selection for Partial Swapping

As in Algorithm 2, prior to determining destination record t'_k to swap the items from t_k, we initially collect all the $t_k \in \mathcal{D}$ which satisfies condition $WR_{t_k,g_j} > MTV_{t_k}$ as the candidate ct_k of destination

record and collect them in a bucket Ct. The following step is to compute the Jaccard similarity coefficient of the t_k with each $ct_k \in Ct$ and obtain the minimum jc, jc_{min} of t_k, using the following equation.

$$jc_{min}(t_k) = \min_{ct_k \in Ct} \frac{(t_k \cap ct_k)}{(t_k \cup ct_k)}. \tag{6}$$

Swapping is performed for such a ct_k denoted by t'_k. Algorithm 2 outputs a set P of pairs (t_k, t'_k) for all $t_k \in \mathcal{D}$, i.e., $P = \{(t_k, t'_k) | t_k \in \mathcal{D}\}$. There are some requisites to determine $ct_k \in Ct$ that can be the best destination record t'_k to swap $i \in g_j$ of t_k. Initially, we check if for each $ct_k \in Ct$ the ct_k has the same tendency category g with t_k. Then, we remove such a ct_k from Ct. This step guarantees that the *true swap* is achieved. The next requisite is that ct_k should result in minimum jc value. The last, ct_k should not contain the same items as in $g_j \in t_k$ to avoid collision which leads to item loss in ct'_k that further it impacts to data utility of an anonymized dataset. Once the ct_k is determined as t'_k , the next phase is selecting $g_j \in t'_k$ which has the highest weight ratio based on Equation (4) and swap the $i \in g_j$ of the t'_k to t_k using Algorithm 3.

Algorithm 2: Procedure of destination record selection for partial swapping.

INPUT : $\mathcal{D}$;
OUTPUT : P ; // P is a set of pairs (t_k, t'_k)
Read $\mathcal{D}$;
Set C_t as an empty set;
forall $t_k \in \mathcal{D}$ **do**
  count MTV_{t_k} of each t_k;
  foreach $g_j \in t_k$ **do**
    count WR_{t_k, g_j} ;
  end
  if WR_{t_k, g_j} *of* $t_k > MTV_{t_k}$ **then**
    $ct_k \leftarrow t_k$;
    append ct_k to Ct;
  end
  foreach $ct_k \in Ct$ **do**
    if ct_k *and* t_k *have the same tendency* **then**
      remove ct_k from Ct;
      calculate $jc(t_k, ct_k)$;
    end
  end
  select ct_k which has the minimum jc out of C_t;
  $t'_k \leftarrow ct_k$;
end

4.4. Procedure of Full Swapping

A record in set-valued databases may contain arbitrary item $i \in I$. Therefore it is necessary to consider a case if individuals have several items from one category only, indicating that the data subject has a very strong tendency to one specific thing. In such a situation we consider to devise an additional algorithm to hide the individual tendency. Still, we emphasize to avoid excessive record distortion.

Before executing Algorithm 4, a set I_{swap} of items are selected as $I_{swap} = \{t_k \in \mathcal{D}\}$. Next, a destination record selection is performed by Algorithm 4 as follows. To achieve the goal, we initially calculate the length of each t_k. For each t_k satisfying $|t_k| > 1$ and containing only one item category $\#g = 1$, append the t_k as ct_k in a bucket Ct. The next step is for each $ct_k \in Ct$ compute jc between the t_k and ct_k. Select the ct_k as the selected destination record t'_k which results in the minimum

jc. Once the t'_k is determined, swap all the items in t_k to t'_k. When the $|Ct| = 0$ which means there are no records satisfying the selection condition of record, full swapping is not possible.

Algorithm 3: Procedure of items swapping.

INPUT : $\mathcal{D}, P, I_{swap}$; // P is a set of pairs (t_k, t'_k)
OUTPUT : $\widetilde{\mathcal{D}}$;
create Buffer B_{t_k} and $B_{t'_k}$;
foreach P **do**
  $B_{t_k} \leftarrow \{i \in g'_j | g'_j \in t_k\}$;
  $B_{t'_k} \leftarrow \{i \in g'_j | g'_j \in t'_k\}$;
  append $B_{t'_k}$ to t_k;
  append B_{t_k} to t'_k;
  save t'_k and t_k to $\widetilde{\mathcal{D}}$;
end
end;

Algorithm 4: Procedure of destination record selection for full swapping.

INPUT : $\mathcal{D}$;
OUTPUT : P ; // P is a set of pairs (t_k, t'_k)
Read $\mathcal{D}$;
Set C_t as an empty set;
forall $t_k \in \mathcal{D}$ **do**
  compute the $|t_k|$;
  if $|t_k| > 1$ ***and*** $\#g \in t_k = 1$ **then**
    $ct_k \leftarrow t_k$;
    append ct_k to Ct;
  end
  foreach $ct_k \in Ct$ **do**
    if ct_k *and* t_k *have the same tendency* **then**
      remove ct_k from Ct;
      calculate $jc(t_k, ct_k)$;
    end
  end
  select ct_k which has the minimum jc out of C_t;
  $t'_k \leftarrow ct_k$;
end

5. Experimental Results and Applicability

In this experiment, we used several real datasets that contained user web click *BMS-WebView*1, *BMS*1 and *BMS-WebView*2, *BMS*2 that has been widely used in knowledge engineering research community. We further tested the proposed method to another real dataset *WD*, containing web access of the Internet users that is generated from *WarpDrive* project [23]. The detail of the datasets' properties is available in Table 3, where $|\mathcal{D}|$ represents dataset size, $\overline{|t|}$ refers to the average item length, while $|i \in \mathcal{D}|$ and $|I|$ are the number of item in dataset and the number of distinct items in dataset, respectively. The experiments were conducted in a computer with processor Intel Core i7 3.4 GHz, RAM 16 GB, and storage 1 TB, under the Windows 7 environment.

Table 3. Properties of $\mathcal{D}$.

Properties	Datasets $\mathcal{D}$		
	BMS1	***BMS2***	***WD***
$\lvert\mathcal{D}\rvert$	5000	5000	797
$\lvert i \in \mathcal{D}\rvert$	12,821	23,530	174,499
$\lvert I\rvert$	264	2717	26,506
$\overline{\lvert t\rvert}$	2.56	4.70	218.94

To evaluate the effectiveness of our proposed technique *PartSwap* we compare it with other techniques i.e., random swapping *RandSwap* which is based on [13], partial suppression *PartSupp* that follows the idea in [7] and *SplitPerson* from [12]. All the methods were implemented in JAVA code with an additional string similarity library in [24] for computing *jc*. We ran the program using all those datasets to obtain anonymized datasets $\widetilde{BMS1}$, $\widetilde{BMS2}$ and $\widetilde{WD}$ and conduct analysis to evaluate the effect of the data anonymization.

5.1. Item Categorization

Prior to applying the methods, initially we create item categorization for the datasets $\mathcal{D}$ like in Figure 1. Ideally, items in a database can be categorized based on their characteristics using a prevalent strategy such as "gates-and-experts" method [25]. However, since the datasets contained only item ID, we follow the idea in [26], that is we create an artificial hierarchal item categorization tree. The tree has a root that represents the most general one i.e., website and it contains 50 categories as its leaves where each leave has equally likely the number of items that randomly selected without any overlapping items among categories from $\mathcal{D}$.

5.2. Data Utility

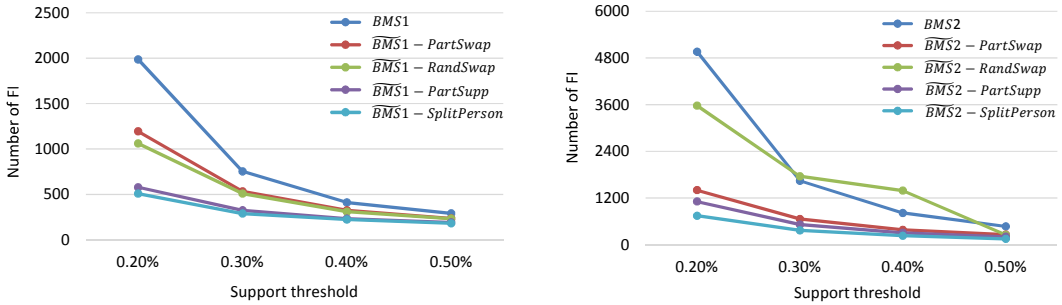
As we have already stated that there is no generic measurement for quantifying data utility. We applied a typical real-world application in set-valued dataset, that is frequent itemset mining. Therefore, we ran FP-Growth algorithm using *spmf* data mining open source software [27] to obtain frequent itemset FI over those three original datasets and their anonymized one.

We determine several support thresholds, ranging from 0.2% to 0.5% to generate FI in *BMS1* and *BMS2* datasets. However, for the *WD* dataset, the support threshold was ranging from 20% to 50% since the dataset has larger number of items and longer tuple length. We evaluated the obtained mining results from the anonymized dataset using our proposed method and that of other methods and compute their similarity to evaluate data accuracy after applying the anonymization methods using the Jaccard similarity measurement in Equation (2).

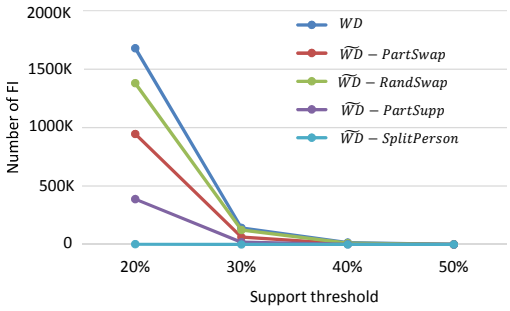
Figure 3 shows that the number of FI from all the anonymized datasets was lower than that of the original one. The decrease in the number of FI in the anonymized datasets mainly due to some items being moved or removed from records, as a result, the structure of records in the anonymized database differs from that of the original one. Since the frequent itemset mining depends on the item sequence, consequently, several itemsets that are considered as frequent itemset in original datasets become no longer frequent in the anonymized datasets and vice versa.

Even so, since our proposed method *PartSwap* only moves items from a certain category which has the highest item frequency in a record to another record, it can successfully maintain the items in the anonymized datasets from loss and minimize record structure distortion. Anonymized databases obtained by *PartSwap* have a considerably higher number of FI compared with those of *PartSupp* and *SplitPerson*. Even though *SplitPerson* does not cause item loss, however, it generates new records in datasets the size of the datasets increases drastically. As the results the number of generated FI under the same support threshold becomes very low. On the other hand, since *RandSwap* also uses swapping

strategy to move items from a record to another, it does not cause item loss. As a result, the method also achieves a significantly higher number of FI.



(**a**) Number of FI in $\widetilde{BMS1}$. (**b**) Number of FI in $\widetilde{BMS2}$.



(**c**) Number of FI in $\widetilde{WD}$.

Figure 3. Comparison of the number of frequent itemset in anonymized datasets and original datasets.

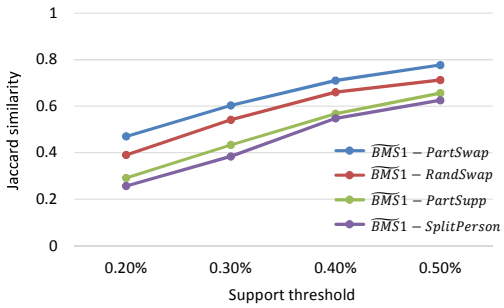
Figure 4 shows that the method always results in higher similarity results. It is indicating that the method results in lower inaccuracy at a record level compared with others. As a result, the obtained FI from the anonymized datasets $\widetilde{BMS1}$-*PartSwap*, $\widetilde{BMS2}$-*PartSwap*, $\widetilde{WD}$-*PartSwap* using the proposed method have the highest number of identical frequent itemset to the original one. Note that we can also observe the following results from our simulation. Even though the proposed method yields the highest number of identical FI, it does not always result in the highest number of FI in the anonymized datasets among all considered methods.

The most prominent feature of our proposed method is that it works very well in a dataset that has a high number of distinct items and a high number of average tuple length. This is because our method selectively determines a pair of record for items swapping and only swap items from a category that has higher item frequency, such that inaccuracy in record level can be minimized. Therefore, we can confirm that due to *PartSwap* results in lower inaccuracy, it successfully preserves higher data utility than other methods.

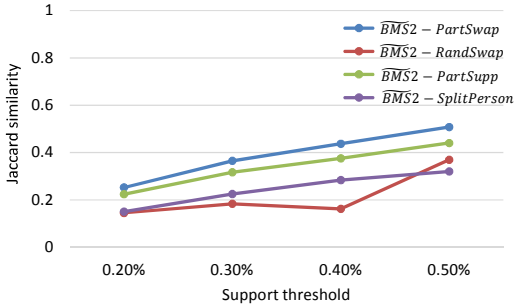
5.3. Data Dissimilarity

Data dissimilarity represents how far the original datasets have been changed to obtain anonymized datasets. To evaluate dissimilarity *Diss*, we use the measurement adopted in [22]. In data swapping technique, the data dissimilarity occurs mainly due to item collision in the swapping process. The collision arises when swapped items already exist in the record destination. Figure 5 shows a comparison of *Diss* values among the anonymized datasets. It represents that our proposed algorithm *PartSwap* achieves lower *Diss* values compared with that of *RandSwap* and *PartSupp*. Such a result can be obtained due to *PartSwap* only swaps items of a category that has the highest item frequency to other items from a different record that has the smallest *jc*. Thus, it can avoid a significant number of item loss in $\widetilde{\mathcal{D}}$ and minimize the *Diss* values. Since *SplitPerson* does not swap or remove

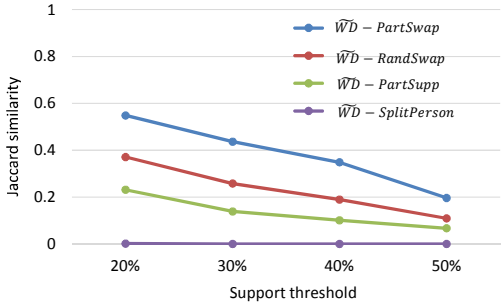
items from datasets, instead, it only splits certain record into several different records the *Diss* value resulted from the method is always zero. On the other hand, *PartSupp* always results in the highest dissimilarity values due to it removes items from the datasets, making the items no longer exist in the anonymized datasets.



(**a**) Similarity of FI in $\widetilde{BMS1}$.

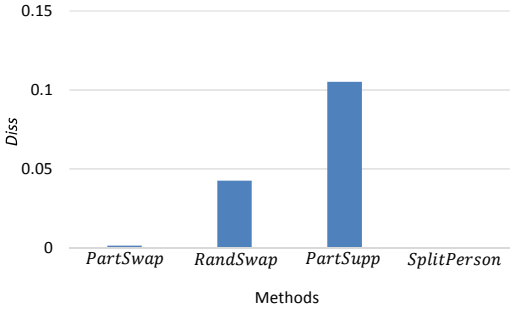


(**b**) Similarity of FI in $\widetilde{BMS2}$.

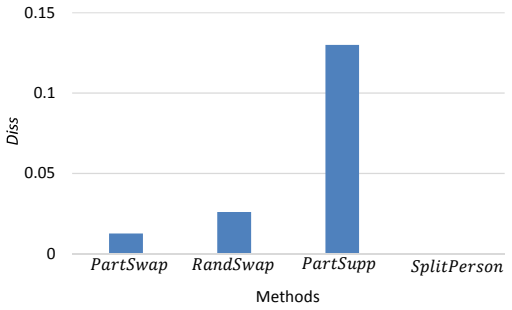


(**c**) Similarity of FI in $\widetilde{WD}$.

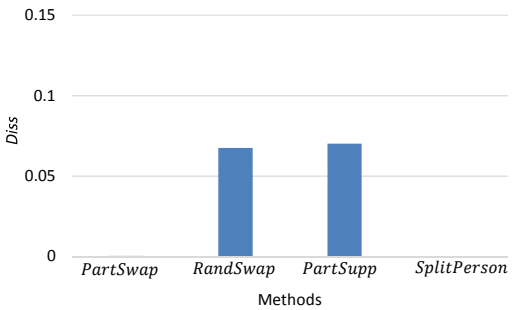
Figure 4. Comparison of the frequent itemset similarity among anonymized datasets to the original datasets.



(**a**) Dissimilarity of $\widetilde{BMS1}$ with $BMS1$.



(**b**) Dissimilarity of $\widetilde{BMS2}$ with $BMS2$.



(**c**) Dissimilarity of $\widetilde{WD}$ with WD.

Figure 5. *Diss* comparison among the anonymized datasets.

5.4. Computational Time

To evaluate the efficiency of our proposed method, we measure computational time for hiding personal tendency. The computational time for $SplitPerson$ has achieved the lowest since the method straightforwardly identifies records in the datasets and split the record containing several personal tendencies into several different records. $PartSupp$ also requires small computational time due to the method only performs item suppression to the records. On the other hand, to successfully hide the personal tendency $RandSwap$ takes the longest computational time since it needs to determine random numbers and assigned them to each record prior to performing items swapping. Prominently, our proposed method requires lower computational time than that of $RandSwap$ since it only needs to calculate jc value between t_k and each t'_k and selects a pair of records that result in the smallest jc. Therefore, the computation time can be reduced.

Figure 6 shows that to generate $\widetilde{BMS1}$ all the methods are able to compute it with quite low computational time since the average tuple length and the number of distinct items in the dataset is small. However, to achieve $\widetilde{BMS2}$ the computational time drastically increases since the datasets have more number of distinct items and also the average length is higher. Even though the number of record in WD dataset is small, it has the largest number of distinct item and the longest average record length so that to achieve $\widetilde{WD}$ all the methods require higher computational time.

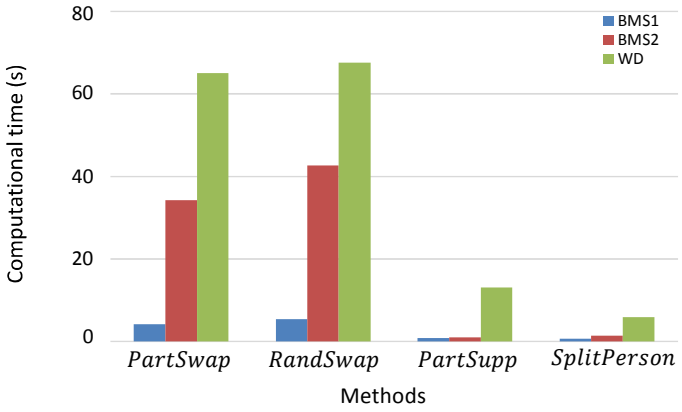


Figure 6. Required computational time to generate $\widetilde{\mathcal{D}}$ among different methods.

5.5. Applicability of the Proposed Method

The proposed algorithm uses weight ratio, WR_{t_k,g_j} and mean tendency value, MTV_{t_k}, both of which can be computed for any set-valued database. After finding category g_j having the maximum weight, the algorithm search for a record as a destination record t'_k for item swapping. Following that, swapping the selected items from t_k and t'_k is executed.

In a certain set-valued database the proposed algorithm cannot perform swapping. To guarantee usefulness of the modified database only categories in a record having the weight ratio higher than MTV_{t_k}, i.e., $WR_{t_k,g_j} > MTV_{t_k}$, are processed for item selection in Section 1. The same inequality is used for selecting destination record t_k in Section 4.3. Suppose $g_1, g_2, .., g_m$ are categories of items in t_k and their item frequency in t_k are $f_1, f_2, ..., f_m$, respectively. From the Equations (4) and (5), the condition that t_k in a database does not satisfy the inequality $WR_{t_k,g_j} > MTV_{t_k}$ becomes $\frac{\sum_{j'=1}^{m} f_{j'}}{m} \geq f_j$ for all $j \in \{1, ..., m\}$, i.e $f_j \leq \frac{\sum_{j'=1 \wedge j' \neq j}^{m} f_{j'}}{m-1}$ for all $j \in \{1, ..., m\}$. Under this condition the Algorithms 1 and 2 cannot output respective I_{t_k} and a pair of P for the t_k. In this situation the swapping cannot be executed for that t_k in the database. A database is not suitable for swapping by the proposed method if most of the t_k in the database cannot be used for swapping because of the condition. For example if all individual records in a database containing categories that have uniform weight, no records can satisfy the condition $WR_{t_k,g_j} > MTV_{t_k}$. As a result the algorithm cannot perform item swapping.

On the other hand, the proposed algorithm needs to compute Jaccard similarity coefficient for all pairs of t_k and t'_k and as the number of records becomes large, its computational time increases. In this sense, the database size should not be too large.

6. Conclusions

In this paper, we have proposed a data anonymization method for hiding personal tendency in set-valued database publication. Our approach successfully hides personal tendency by exploiting Jaccard similarity coefficient among records in datasets and swapping items from a certain category in a record to other items from different category in a targeted record.

In addition, by swapping only items from a category that has the highest frequency in a record the distortion of record structure in the anonymized database can be minimized, as a result, it successfully preserves higher data utility and results in smaller dissimilarity values.

Even though the swapping techniques always result in inaccuracy at a record level, it can bring advantage for data recipients since they can obtain all information about the items even if the database has been anonymized. The swapping technique is also beneficial for certain application such as recommender systems to increase user serendipity as a solution to avoid overspecialization and broaden user preferences.

In terms of the computational time, our proposed method requires considerably lower computational time compared with that of existing swapping technique. The main reason is that the proposed method measures the Jaccard similarity coefficient of each pair of records and takes the pair that has the smallest similarity coefficient value for items swapping. It is also important to note that the size, the number of distinct items and the record length of databases greatly affect the computational time.

Author Contributions: D.G. devised the problem and reviewed related work, designed the solution and programmed the simulation application, wrote the original draft, and reviewed and edited the draft. M.M. designed idea, gave improvement, reviewed and edited the draft as well as supervised the overall quality of the article.

Acknowledgments: The research results have been partially achieved by WarpDrive: Web-based Attack Response with Practical and Deployable Research InitiatiVE, the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN. The first author thanks to BUDI-LN scholarship from Indonesia Endowment Fund for Education (LPDP) and Ministry of Research, Technology and Higher Education of Republic of Indonesia (RISTEKDIKTI).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, H.; Sarathy, R.; Xu, H. The Role of Affect and Cognition on Online Consumers' Decision to Disclose Personal Information to Unfamiliar Online Vendors. *Decis. Support Syst.* **2011**, *51*, 434–445. [CrossRef]
2. Sweeney, L. K-anonymity: A Model For Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
3. Terrovitis, M.; Mamoulis, N.; Kalnis, P. Privacy-preserving anonymization of set-valued data. *Proc. VLDB Endow.* **2008**, *1*, 115–125. [CrossRef]
4. Xu, Y.; Fung, B.C.M.; Wang, K.; Fu, A.W.C.; Pei, J. Publishing Sensitive Transactions for Itemset Utility. In Proceedings of the 2008 Eighth ICDM '08 IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE Computer Society: Washington, DC, USA, 2008; pp. 1109–1114.
5. Liu, J.Q. Publishing set-valued data against realistic adversaries. *J. Comput. Sci. Technol.* **2012**, *27*, 24–36. [CrossRef]
6. LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Incognito: Efficient Full-Domain K-Anonymity. In *SIGMOD Conference*; Özcan, F., Ed.; ACM: New York, NY, USA, 2005; pp. 49–60.
7. Jia, X.; Pan, C.; Xu, X.; Zhu, K.Q.; Lo, E. ρ-uncertainty Anonymization by Partial Suppression. In *International Conference on Database Systems for Advanced Applications*; Springer: Berlin, Germany, 2014; pp. 188–202.

8. Ghinita, G.; Kalnis, P.; Tao, Y. Anonymous Publication of Sensitive Transactional Data. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 161–174. [CrossRef]
9. Reiss, S.P.; Post, M.J.; Dalenius, T. Non-reversible Privacy Transformations. In Proceedings of the 1st ACM PODS '82 SIGACT-SIGMOD Symposium on Principles of Database Systems, Los Angeles, CA, USA, 29–31 March 1982; ACM: New York, NY, USA, 1982; pp. 139–146.
10. Evfimievski, A.; Gehrke, J.; Srikant, R. Limiting privacy breaches in privacy preserving data mining. *Pods* **2003**, 211–222. [CrossRef]
11. Farnan, N.L.; Lee, A.J.; Chrysanthis, P.K.; Yu, T. Don't Reveal My Intension: Protecting User Privacy Using Declarative Preferences during Distributed Query Processing. In *Computer Security–ESORICS 2011*; Atluri, V., Diaz, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 628–647.
12. Adar, E. User 4xxxxx9: Anonymizing query logs. In Proceedings of the of Query Log Analysis Workshop, International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007.
13. Reiss, S.P. Practical Data-swapping: The First Steps. *ACM Trans. Database Syst.* **1984**, *9*, 20–37. [CrossRef]
14. Kotkov, D.; Wang, S.; Veijalainen, J. A Survey of Serendipity in Recommender Systems. *Know.-Based Syst.* **2016**, *111*, 180–192. [CrossRef]
15. Domingo-Ferrer, J.; Torra, V. *Theory and Practical Applications for Statistical Agencies, Chapter A Quantitative Comparison of Disclosure Control Methods for Microdata, Confidentiality, Disclosure and Data Access*; Elsevier: Amsterdam, The Netherlands, 2002.
16. Nin, J.; Herranz, J.; Torra, V. Rethinking rank swapping to decrease disclosure risk. *Data Knowl. Eng.* **2008**, *64*, 346–364. [CrossRef]
17. Hasan, A.S.M.T.; Jiang, Q.; Luo, J.; Li, C.; Chen, L. An Effective Value Swapping Method for Privacy Preserving Data Publishing. *Secur. Commun. Netw.* **2016**, *9*, 3219–3228. [CrossRef]
18. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-diversity: Privacy Beyond K-anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*. [CrossRef]
19. Domingo-Ferrer, J.; Torra, V. Disclosure Risk Assessment in Statistical Data Protection. *J. Comput. Appl. Math.* **2004**, *164-165*, 285–293. [CrossRef]
20. Nakagawa, T.; Arai, H.; Nakagawa, H. Personalized Anonymization for Set-Valued Data by Partial Suppression. *Trans. Data Priv.* **2018**, *11*, 219–237.
21. Han, J.; Pei, J.; Yin, Y. Mining Frequent Patterns Without Candidate Generation. *SIGMOD Rec.* **2000**, *29*, 1–12. [CrossRef]
22. Oliveira, S.R.M.; Zaiane, O.R. Privacy Preserving Frequent Itemset Mining. In Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining (PSDM 2002), Maebashi City, Japan, 9 December 2002; Clifton, C., Estivill-Castro, V., Eds.; ACS: Maebashi City, Japan, 2002; Volume 14, pp. 43–54.
23. WarpDrive-Project. Web-Based Attack Response with Practical and Deployable Research Initiative. 2018. Available online: https://warpdrive-project.jp/ (accessed on 1 June 2018).
24. Debatty, T. Java String Similarity. 2014. Available online: https://github.com/tdebatty/java-string-similarity (accessed on 14 October 2018).
25. Shen, D.; Ruvini, J.D.; Somaiya, M.; Sundaresan, N. Item Categorization in the e-Commerce Domain. In Proceedings of the 20th CIKM '11 ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; ACM: New York, NY, USA, 2011; pp. 1921–1924.
26. He, Y.; Naughton, J.F. Anonymization of Set-Valued Data via Top-Down, Local Generalization. *Proc. VLDB Endow.* **2009**, *2*, 934–945. [CrossRef]
27. Fournier-Viger, P.; Lin, J.C.; Gomariz, A.; Gueniche, T.; Soltani, A.; Deng, Z.; Lam, H.T. The SPMF Open-Source Data Mining Library Version 2. In Proceedings of the Part III Machine Learning and Knowledge Discovery in Databases–European Conference, ECML PKDD 2016, Riva del Garda, Italy, 19–23 September 2016; pp. 36–40.